Simulazione dell'evoluzione di un'epidemia attraverso il modello SIR e il modello dell'automa cellulare

Alessio Belshi, Andrea Doddi, Andrea Fraternali, Francesco Bolognesi

Abstract

Con il seguente progetto si vuole fornire un utile strumento per simulare l'andamento di un qualsiasi epidemia, anche se con alcune semplificazioni. Ilprogetto è suddiviso in due parti. Nella prima l'evoluzione dell'epidemia è stata simulata attraverso il modello SIR. Nella seconda parte, invece, si è utilizzato il modello dell'automa cellulare.
I risultati ottenuti sono via via più compatibili al crescere del numero di persone coinvolte nell'epidemia.
Per la condivisione del lavoro, si è utilizzato *git*. Il progetto è perciò reperibile qui.

Indice

1 Introduzione

1.1 Modello SIR

Il SIR è un modello matematico utilizzato per monitorare e prevedere lo sviluppo di un epidemia. La popolazione coinvolta nel fenomeno viene suddivisa in tre classi: quella dei suscettibili (S), cioè coloro che possono contrarre la malattia; quella degli infetti (I) e quella dei rimossi (R), che comprendono i guariti e i deceduti. L'evoluzione temporale dei tre parametri S, I ed R è descritta da altrettante equazioni differenziali:

$$\frac{dS}{dt} = -\beta \frac{S}{N} I \tag{1}$$

$$\frac{dI}{dt} = \beta \frac{S}{N} - \gamma I \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

dove N è la popolazione totale coinvolta, mentre i parametri β e γ sono compresi tra 0 e 1 e quantificano rispettivamente la probabilità di contagio a seguito di contatto tra suscettibile e infetto, e la probabilità di guarigione (o di morte).
Affinchè l'epidemia abbia inizio si deve verificare la condizione $\frac{\beta}{\gamma} > 1$. Inoltre, l'epidemia è in fase di espansione quando $\frac{S}{N} > \frac{\gamma}{\beta}$, e in fase di contrazione nel caso opposto. Per l'implementazione del programma si sono discretizzate le equazioni ponendo $\Delta t = 1$:

$$S_i = S_0 + S_{i-1} - \beta \frac{S_{i-1}}{N} I_{i-1} \tag{4}$$

$$I_i = I_{i-1} + \beta \frac{S_{i-1}}{N} - \gamma I_{i-1} \tag{5}$$

$$R_i = R_{i-1} + \gamma I_{i-1} \tag{6}$$

Inoltre i parametri β e γ sono stati considerati costanti durante l'evoluzione dell'epidemia.

1.2 Automa cellulare

L'automa cellulare consiste in una griglia costituita da un numero finito di celle. Ogni cella assume uno stato contraddistinto da un determinato colore. L'evoluzione dello stato di ogni cella dipende dallo stato delle celle limitrofe secondo una regola prestabilita. Per simulare l'epidemia si è deciso che ogni cella rappresenti una persona, il cui stato può essere quello di suscettibile, infetto o rimosso, come nel modello SIR.

2 Struttura del codice

Il progetto è suddiviso in due cartelle: la cartella *sir_model* contiene il codice relativo al modello SIR, mentre la cartella *cell_automaton* quello relativo all'automa cellulare.

2.1 Modello SIR

I parametri presi in input dal programma sono:

- la durata in giorni della simulazione
- i parametri S, I ed R al giorno 0
- I parametri β e γ

I giorni e i parametri S, I ed R sono di tipo *int* e devono essere positivi, mentre β e γ sono di tipo *double* e devono essere compresi tra 0 e 1.
L'implementazione utilizza un file main (*main.sir.cpp*), che si occupa prevalentemente della parte grafica, un file sorgente (*epidemic.cpp*) e un file header (*epidemic.hpp*) in cui è definita la classe *Epidemic*, che gestisce l'evoluzione dell'epidemia, insieme con i suoi metodi. In particolare, il metodo *evolve* calcola lo stato successivo dell'epidemia, basandosi sulle equazioni (4) (5) e (6) mentre il metodo rounding_int si occupa di mantenere costante la somma :

$$S + I + R = N \tag{7}$$

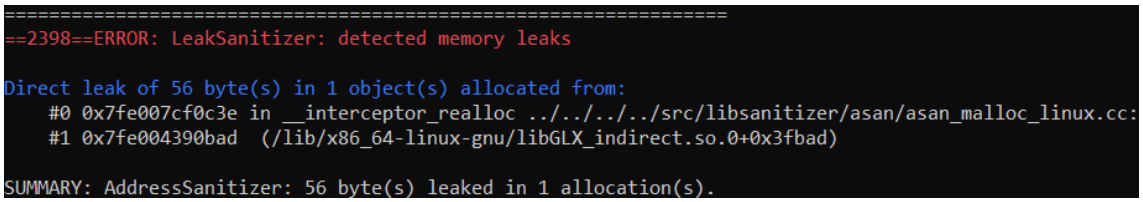
$$S, I, R \in N \tag{8}$$

Gli stati giornalieri vengono riportati in un grafico disegnato in SFML e costruito con la classe *Graph*, definita nei file *graph.hpp* e *graph.cpp*. L'andamento di ogni parametro è rappresentato da una serie di punti (una

curva discretizzata).
Il file *times.ttf* contiene il font utilizzato per le scritte a video.
Per compilare utilizzare all'interno della cartella *sir_model*:

g++ -Wall -Wextra -fsanitize=address graph.cpp main.sir.cpp epidemic.cpp -lsfml-window -lsfml-system -lsfml-graphics -O3

Per eseguire, utilizzare *./a.out*.
È possibile che si incontri un messaggio di errore del tipo:



Si tratta di un problema all'interno della libreria *libGLX_indirect.so*, che può essere ignorato.

2.2 Automa cellulare

I parametri presi in input sono:

- la larghezza e l'altezza della griglia (il prodotto corrisponde alla popolazione totale)
- il numero di infetti e di rimossi iniziali
- i parametri β e γ
- la durata in giorni della simulazione

Le condizioni sui dati in input sono le stesse descritte per il modello SIR. L'implementazione è divisa tra un file main (*main.autom.cpp*), un file sorgente e un header (rispettivamente *automaton.cpp* e *automaton.hpp*) che gestiscono la classe omonima e i file *graph.hpp* e *graph.cpp* utilizzati anche per il modello SIR.
La classe *Automaton* implementa la griglia dell'automa cellulare come un vettore di celle, dove ogni cella è un *enum* che può essere S, I o R. I metodi centrali sono i metodi *set* e *evolve*. Il primo assegna ad una determinata cella un preciso stato, il secondo esegue un'evoluzione dell'automa. Si è scelto di

far passare ogni cella S ad I con una probabilità $\frac{n}{8}\beta$, dove n è il numero di celle infette tra le 8 adiacenti (valore ottenuto attraverso il metodo privato *check*). In questo modo la probabilità di contagiarsi è il prodotto tra la probabilità di incontrare un infetto e la probabilità di contrarre il virus dopo l'incontro. Le celle I, invece, diventano R con una probabilità γ.
Ai bordi della griglia le celle possono essere considerate a contatto con quelle del bordo opposto (*effetto pacman*).
Simultaneamente all'evoluzione della griglia, viene disegnato un grafico aggiungendo giorno per giorno i punti che rappresentano i valori di S, I ed R. Il file main prende in input i parametri richiesti e fa evolvere la griglia dinamicamente: ogni giro del *game loop* è un'evoluzione dell'automa cellulare calcolata attraverso la classe *Automaton* e rappresentata graficamente, in SFML, una griglia colorata. Simultaneamente all'evoluzione della griglia, viene disegnato un grafico aggiungendo giorno per giorno i punti che rappresentano i valori di S, I, R.
Troviamo inoltre il già citato file *times.ttf* per il font. Per compilare, all'interno della cartella *cell_automaton* usare l'opzione:

*g++ -Wall -Wextra -fsanitize=address *.cpp -lsfml-window -lsfml-system -lsfml-graphics -O3*

Anche in questo caso si può riscontrare il problema riportato sopra e ugualmente lo si può ignorare.
Eseguire con *./a.out*.

3 Strategie di testing

3.1 Modello SIR

I test implementati per verificare il corretto funzionamento del programma sono contenuti nel file *epidemic.test.cpp*. Vi è un unico test_case, suddiviso in più subcase. I subcase sono caratterizzati da diversi valori inziali dei parametri S, I, R, β e γ, e dai valori attesi secondo il modello SIR. Sono stati considerati i casi più significativi quali quelli in cui uno dei parametri è uguale a 0, quelli in cui l'epidemia è in espansione o in contrazione e quello in cui l'epidemia si arresta. Inoltre, si è implementato un test per verificare che, quando i valori di due parametri terminano contemporanemaente con un 5 come decimale, il programma ne approssimi uno per difetto e l'altro per eccesso, per evitare una sovrastima di N.
Per eseguire i test, compilare con l'opzione:

g++ -Wall -Wextra -fsanitize=address epidemic.test.cpp epidemic.cpp -O3

ed eseguire con *./a.out*.

4 Risultati

L'evoluzione dell'epidemia risulta ovviamente differente nei due modelli, anche a parità di configurazione iniziale. Questo perchè le equazioni del modello SIR trattano il problema in maniera statistica, mentre l'automa cellulare prende in considerazione ogni singola cella, che interagisce solamente con quelle attorno. Tuttavia, l'automa cellulare posiziona casualmente le celle inizialmente infette e rimosse, mentre la probabilità di infettarsi è calcolata combinando la probabilità di incontro con un infetto e quella di contagio a seguito dell'incontro.
Questa scelta riflette il funzionamento stocastico del modello SIR, con l'unica differenza che lì si immagina che ogni persona interagisca con tutta la popolazione.
Ciò fa sì che ci sia una discreta differenza quantitativa tra i due modelli, nonostante l'andamento qualitativo sia simile al crescere del numero di giorni e delle persone coinvolte.